Exhibit B

Efficient Mixing of Sequential Prefetches with Random Access Data in a Preexisting LRU Cache

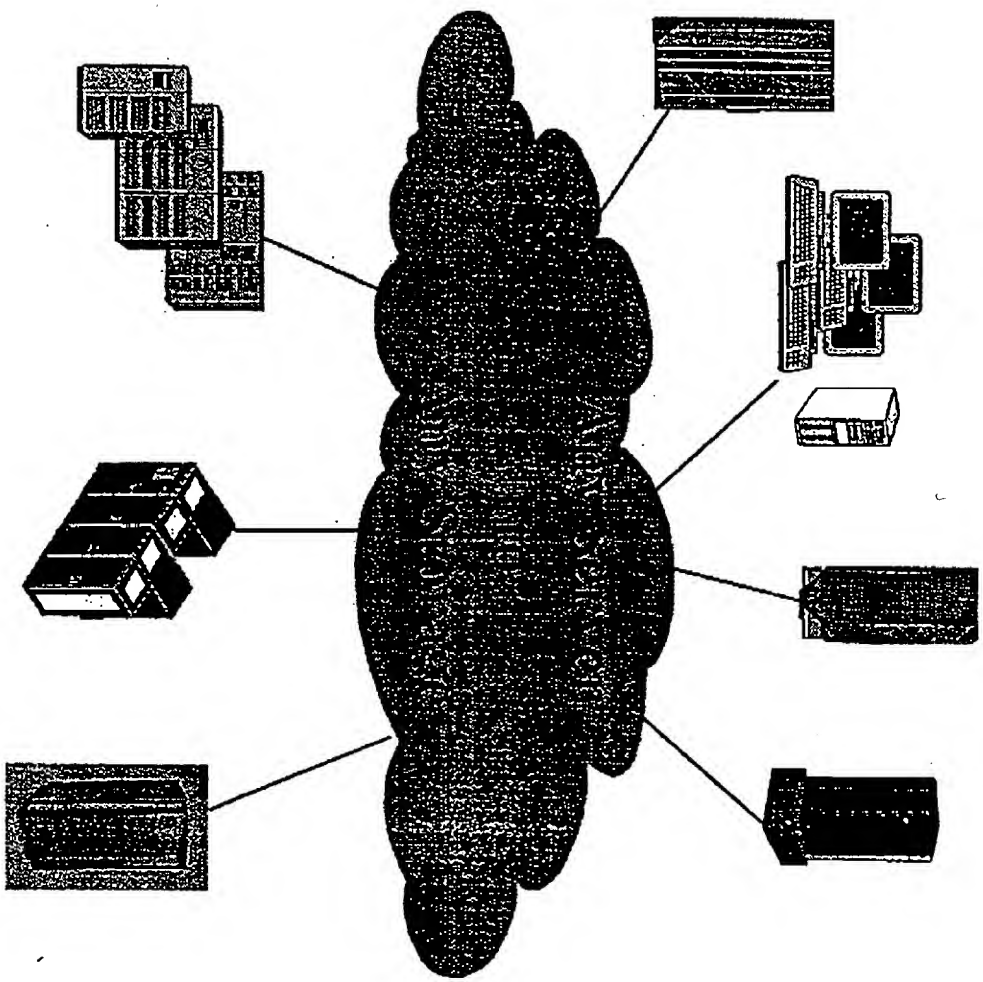
Patent Review Board

IBM Confidential

Efficient Prefetching w/LRU...

- Title:
  - Efficient Mixing of Sequential Prefetches with Random Access Data in a Preexisting LRU Cache.
- Product Name: none (several candidates: CF, SAN)
- First Date Disclosed: Has not been disclosed.
  -
- Problem Solved: Prefetch optimization in a black-box LRU caching system.

Example use of the process in a SAN environment



Problem

- LRU cache logic sometimes inaccessible or undesirable to alter.
- LRU cache often accessed by external hosts or multiple hosts.
- Prefetched data is ideal for sequential accesses, which works poorly with LRU.
- Identifying prefetch candidates difficult.

Solution

- Solution is to:
  - Use model of prefetching effects to make decisions.
  - Estimate single ref. residency time.

Efficient Prefetching w/LRU...

- Benefit over prior art:
- Competitors:
  - ►
- Will others want to use it?
- Discoverable?
  - ► how easily?
- Alternatives?

- Improved identification of prefetch candidates.
- Modification of LRU not required.
- Companies trying to preload selected elements in an LRU cache.
- Consider a SAN with a cached control unit to preload into.
- I/O requests in addition to workload together with a moving prefetch criteria.
- Static prefetch determination.

The process for efficient prefetching w/LRU

- Determine cache size.
- Periodically fetch Hit Ratio, cached I/O rate and estimate the SRRT.
- For each I/O check the model's buffer, use previous element's sequential count.
- If above a dynamic threshold, prefetch.

The process for efficient prefetching w/LRU *(continued)*

- Load I/O requests + prefetches into model's buffer. Update LRU position on hit. Discard on overflow.
- Keep track of overflow point for each of multiple prefetch threshold criteria, counting prefetched "hits" for each criteria.
- If alternate prefetch criteria value is better, then consider adjustment.

Description of Invention

- Inputs:
  - Cache Size
  - periodic:
    - Hit Ratio feedback
    - Cached I/O Rate feedback
- Output:
  - Prefetch candidates identified
  - Optimal prefetch threshold identified
  - Value of prefetching quantifiable